Algebraic constructions of LDPC codes with no short cycles.

Ted Hurley[*]   Paul McEvoy [†]   Jakub Wenus [‡]

Abstract

An algebraic group ring method for constructing codes with no short cycles in the check matrix is derived. It is shown that the matrix of a group ring element has no short cycles if and only if the collection of group differences of this element has no repeats. When applied to elements in the group ring with small support this gives a general method for constructing and analysing low density parity check (LDPC) codes with no short cycles from group rings. Examples of LDPC codes with no short cycles are constructed from group ring elements and these are simulated and compared with known LDPC codes, including those adopted for wireless standards.

1 Introduction

An LDPC code is a code where the check matrix has only a small number of non-zero elements in each row and column. These were introduced by Gallager in [1], expanded further by Tanner in [14], and rediscovered [8, 9] and expanded on by MacKay and Neal; details can now be found in [6]. Structured LDPC codes usually use various types of combinatorial objects such as designs or algebraic geometry. LDPC codes have often been produced by randomised techniques, but there has been recent activity in the area of algebraic constructions [13, 12, 15]. Having no short cycles in the (Tanner) graph of the check matrix of an LDPC code has been shown to dramatically improve the performance of the code. Short cycles in an LDPC code deteriorate the performance of the decoding algorithms and having no short cycles may in effect increase the distances of such codes.

Here a group ring method for the construction and analysis of LDPC codes with no short cycles is presented.

1.1 Group ring method

Cyclic and related codes owe much of their structure and properties because they occur as ideals or modules within a cyclic group ring. So also general group rings may be used to construct, analyse and give structure to many other types of codes and may be used to construct codes of a particular type or with a particular property. Group ring module codes are obtained from either zero-divisors or units in a group ring RG as described in [5] or [4]. Thus elements $u, v \in RG$ are considered with either $uv = 0$ (zero divisors) or $uv = 1$ (units), where 1 denotes the identity of RG, and codes are derived therefrom.

The unit-derived group ring code method is particularly useful and has great flexibility while still retaining much of the algebraic structure. This method is employed here to directly and algebraically construct low density parity check (LDPC) codes with no short cycles in their (Tanner) graphs. Zero-divisors may also be used but the unit-derived method has advantages and has less theoretical complications.

---

[*]National University of Ireland, Galway, Ireland. ted.hurley@nuigalway.ie

[†]TechnologyFromIdeas, Old Kilmeaden Road, Waterford, Ireland. paul.mcevoy@technologyfromideas.com

[‡]TechnologyFromIdeas, Old Kilmeaden Road, Waterford, Ireland. jakub.wenus@technologyfromideas.com

Using an injection $\phi : RG \rightarrow R_{n \times n}$, as for example in [3], from the group ring RG with $|G| = n$ into the ring of $n \times n$ matrices over R, corresponding matrix codes are obtained from the group ring codes. In this injection the notation $\phi(v) = V$ is used so that the capital letter V in the matrix ring corresponds to the lower case letter v in the group ring.

Thus we are lead to consider $u, v \in RG$ with $uv = 1 = vu$ from which the codes are defined. Certain rows are chosen from the matrix U of the unit u to form the generator matrix of a code and the corresponding columns are deleted from the matrix V of the inverse v of u to form the check matrix of this code.

If v has short support then its corresponding matrix V has only a small number of elements in each row and column and is thus low density. 'Short support' of a group ring element v means that only a small number (compared to the size of the group) of the coefficients in v are non-zero. If the check matrix of a group ring code is derived from a group ring check element v with small support then the resulting code will be an LDPC code.

It is determined here where precisely the short cycles can occur in the matrix of a group ring element. It is then easy to construct group ring elements, and group ring elements of small support, which will have no short cycles anywhere in their matrices. Using group ring elements with small support and with no short cycles in their matrices, LDPC codes with no short cycles are constructed.

Thus LDPC codes with no short cycles can be constructed by the following algebraic group ring method:

1. Construct u, v is a group ring RG with $uv = 1$ and such that v has small support compared to the size $|G|$ of the group G. These can be constructed with some property in mind. Units in group rings abound and are easy to construct.

2. Decide on the rate $\frac{r}{|G|}$ of the code required. This is often decided by reference to u, v and their structures.

3. Choose r rows of U with which to construct the generator matrix and delete the corresponding r columns of V to form the check matrix. This gives a rate $\frac{r}{|G|}$ code.

4. If V has no short cycles at all, which can be ensured by Theorem 2.1 below, then any choice of columns of V and consequent choice of rows of U will give an LDPC code with no short cycles.

5. If V has short cycles, it may be possible to avoid these by deleting appropriate columns and still obtain an LDPC code with no short cycles.

It is of obviously easier to ensure there are no short cycles in the resulting LDPC codes if the original (check) element v from which the code is constructed has no short cycles at all in its matrix V. This can be ensured in the construction of v by Theorem 2.1 below.

Another advantage of codes derived from units is that there is a huge choice of columns from which to form a check matrix of a code. For example suppose from a unit of size 1000×1000 with no short cycles and low density a $(1000, 500)$ code is required. There are $\binom{1000}{500}$, which is of the order of 2^{500}, choices from the 1000×1000 matrix with which to form the code and each code is low density and has no short cycles. From the nature of the independence of any set of rows (or columns) in a unit (= non-singular) matrix each code derived is different.

One could thus envisage a hybrid whereby a random construction is performed within the parameters of an algebraic construction.

1.2 Examples and simulations

Examples, simulations and comparisons are given in Section 3. These compare extremely well with existing LDPC codes and in many cases outstrip them. An example is also given of a unit from which 10 random LDPC codes of rate 1/2 are constructed. These are then simulated and all of them performed well. The

potential applications of having random LDPC codes with no short cycles derived from a single unit and all performing well are obvious. In addition comparisons of Bit Error Rate (BER) and Block Error Rate (BLER) performance of LDPC codes defined in the 802.11n & 802.16e standard with equivalent codes generated by the present method are given in Section 3.6.

BER is not everything and often fast and power-efficient coding is more important than performance. The group ring method for LDPC codes needs only a relatively few initial parameters and can re-create the matrix line-by-line without the need to store the whole structure in memory. The method has thus in addition applications where low power and low storage are requirements.

1.3   Notation

RG denotes the group ring of the group G over the ring R; when R is a field, RG is often referred to as a *group algebra*. No deep knowledge of group rings is required but familiarity with the ideas of *units, zero-divisors* in rings is assumed. For further information on group rings see [11]. C_n denotes the cyclic group of order n and $H \times K$ denotes the direct product of groups H, K.

The words 'graph' and 'short cycles' is used but as now explained no knowledge of graph theory is required and the problem of avoiding short cycles reduces to looking at a property of matrices.

For any matrix $H = (h_{ij})$ the Tanner graph [14] of H is a bipartite graph $K = V_1 \cup V_2$ where V_1 has one vertex for each row of H and V_2 has one vertex for column in H and there is an edge between two vertices i, j exactly when $h_{ij} \neq 0$. A short cycle in the (Tanner) graph of a matrix is a cycle of length 4.

Thus a matrix has no short cycles in its graph if and only the intersection of positions in which two columns have non-zero values is at most 1. This definition is used when considering the absence or otherwise of short cycles and thus no deep graph theory is involved.

2   Avoiding short cycles

Avoiding short cycles in the (Tanner) graph of the check matrix of a code is important, particularly for LDPC (low density parity check) codes.

Specifically here, necessary and sufficient conditions are given on the group ring element v in terms of the group elements with non-zero coefficients occurring in it so that its corresponding matrix V has no short cycles. A mathematical proof is given.

Some special cases, such as when G is cyclic or abelian, of the general result are easier to describe and useful in practice and these are used as examples and illustrations of the general results.

2.1   Collections of differences, special case

Collections of differences are usually defined with respect to a set a non-negative integers, see for example [2]. Collections of group differences are defined in Section 2.3 and the collections of (integer) differences are special cases of these when the group is a cyclic group.

The integer definition is recapped here and used to give examples of the general definition.

Let $S = \{i_1, i_2, \ldots, i_r\}$ be a set of non-negative unequal integers and n an integer with $n > i_j$ for all $j = 1, 2, \ldots, r$.

Then the *collection of cyclic differences of S* mod n is defined by $DS(n) = \{i_j - i_k \mod n | 1 \leq j, k \leq r, j \neq k\}$. This collection has possibly repeated elements.

For example if $S = \{1, 3, 7, 8\}$ and $n = 12$ then $DS(12) = \left\{ \begin{array}{ccc} 2 & 6 & 7 \\ & 4 & 5 \\ & & 1 \\ 10 & 6 & 5 \\ & 8 & 7 \\ & & 11 \end{array} \right\} = \{2, 6, 7, 4, 5, 1, 10, 6, 5, 8, 7, 11\}.$

In this case $6, 7, 5$ occur twice.

If $|S| = r$ then counting repeats $|DS(n)| = r(r-1)$.

2.2 Cyclic group ring differences

Consider the group ring RC_n where C_n is the cyclic group of order n generated by g. Suppose $u = \alpha_{i_1} g^{i_1} + \alpha_{i_2} g^{i_2} + \ldots + \alpha_{i_r} g^{i_r} \in RC_n$ with $\alpha_{i_j} \neq 0$ (and $0 \leq i_j < n$).

For each g^i, g^j in u with non-zero coefficients form $g^i g^{-j}, g^j g^{-i}$ and define $DS(u)$ to be the collection of all such $g^i g^{-j}, g^j g^{-i}$.

Set $S = \{i_1, i_2, \ldots, i_r\}$ and define the collection of cyclic differences $DS(n)$ as above. It is clear that $DS(n)$ and $DS(u)$ are equivalent, the only difference being in the notation used. The proof of the following theorem is a direct corollary of the more general Theorem 2.2 below.

Theorem 2.1 U has no 4-cycles in its graph if and only if $DS(u)$ has no repeated elements.

2.2.1 Example

Set $u = 1 + g + g^3 + g^7$ in $\mathbb{Z}_2 C_{15}$. The collection of differences is formed from $\{0, 1, 3, 7\}$ and is thus

$DS(u) = \{1, 3, 7, 2, 6, 4, 14, 12, 8, 13, 9, 11\}$ which has no repeats. Hence the matrix formed from u, which is circulant in this case, has no short cycles.

Set $u = 1 + g + g^3 + g^7$ in $\mathbb{Z}_2 C_{13}$.

The collection of differences formed from $\{0, 1, 3, 7\}$ is $\{1, 3, 7, 2, 6, 4, 12, 10, 6, 11, 7, 9\}$ and has repeats $6, 7$.

Thus the matrix formed from u has short cycles – but we can identify where they occur.

2.3 Collection of differences in a general group ring

Let RG denote the group ring of the group G over the ring R. Let G be listed by $G = \{g_1, g_2, \ldots, g_n\}$.

Let $u = \displaystyle\sum_{i=1}^{n} \alpha_i g_i$ in RG.

For each (distinct) pair g_i, g_j occurring in u with non-zero coefficients, form the (group) *differences* $g_i g_j^{-1}, g_j g_i^{-1}$. Then *the collection of difference* of u, $DS(u)$, consists of all such differences. Thus:

$DS(u) = \{g_i g_j^{-1}, g_j g_i^{-1} | g_i \in G, g_j \in G, i \neq j, \alpha_i \neq 0, \alpha_j \neq 0\}.$

Note that the collection of differences of u consists of group elements and for each $g, h, g \neq h$, occurring with non-zero coefficients in u both gh^{-1} and its inverse hg^{-1} are formed as part of the collection of differences.

Theorem 2.2 *The matrix U has no short cycles in its graph if and only if $DS(u)$ has no repeated (group) elements.*

Proof: The rows of U correspond in order to $ug_i, i = 1, \ldots, n$, see [5].

Then U has a 4-cycle

$\Longleftrightarrow$

for some $i \neq j$ and some $k \neq l$, the coefficients of g_m, g_l, in ug_i and ug_j are nonzero.

$\Longleftrightarrow$

$ug_i = ... + \alpha g_k + \beta g_l + ...$

and

$ug_j = ... + \alpha_1 g_k + \beta_1 g_l + ...$

$\Longleftrightarrow$

$u = ... + \alpha g_k g_i^{-1} + \beta g_l g_i^{-1} + ...$

and

$u = ... + \alpha_1 g_k g_j^{-1} + \beta_1 g_l g_j^{-1} + ...$

$\Longleftrightarrow$

$DS(u)$ contains both $g_k g_i^{-1} g_i g_l^{-1} = g_k g_l^{-1}$ and $g_k g_j^{-1} g_l^{-1} = g_k g_l^{-1}$.

This happens if and only if $DS(u)$ has a repeated element. $\qquad\qquad\square$

2.4 Repeated elements

Suppose now u is such that $DS(u)$ has repeated elements.

Hence $u = ... + \alpha_m g_m + \alpha_r g_r + \alpha_p g_p + \alpha_q g_q + ...$, where the displayed α_i are not zero, so that $g_m g_r^{-1} = g_p g_q^{-1}$. The elements causing a short cycle are displayed and note that the elements g_m, g_r, g_p, g_q are not necessarily in the order of the listing of G.

Since we are interested in the graph of the element and thus in the non-zero coefficients, replace a non-zero coefficient by the coefficient 1. Thus write $u = ... + g_m + g_r + g_p + g_q + ...$ so that $g_m g_r^{-1} = g_p g_q^{-1}$.

Include the case where one p, q could be one of m, r in which case it should not be listed in the expression for u.

Then $ug_m^{-1} g_p = .. + g_p + g_r g_m^{-1} g_p. + ... = ... + g^p + g^q + ..$

and $ug_p^{-1} g_m = ... + g_m + g_q g_p^{-1} g_m = ... + g_m + g_r + ...$

(Note that $ug_m^{-1} g_p = ug_r^{-1} g_q$ and $ug_p^{-1} g_m = ug_q^{-1} g_r$)

Thus to avoid short cycles, do not use the row determined by $g_m^{-1} g_p$ or the row determined by $g_p^{-1} g_m$ in U if using the first row or in general if g_i row occurs then $g_i g_m^{-1} g_p$, and $g_i g_p^{-1} g_m$ rows must not occur.

Similarly when $DS(u)$ has repeated elements by avoiding certain columns in U, it is possible to finish up with a matrix without short cycles.

2.5 Special group cases

The special case when $G = C_n$ was dealt with in Section 2.2.

Let $G = C_n \times C_m$ be the direct product of cyclic groups C_n, C_m generated by g, h respectively. These groups are particularly useful in practice.

List the elements of G by $\{1, g, g^2, \ldots, g^{n-1}, h, hg, hg^2, \ldots, hg^{n-1}, \ldots, h^{m-1}, h^{m-1}g, \ldots, h^{m-1}g^{n-1}\}$.

Then every element in RG is of the form $u = a_0 + ha_1 + \ldots + h^{m-1}a_{m-1}$ with each $a_i \in C_n$. The collection of differences of u is easy to determine and elements with no repeats in their collection of differences are thus easy to construct.

Relative to this listing the matrix of an element in RG is a circulant-by-circulant matrix of size $mn \times mn$,

[3].

Another particularly useful group which is relatively easy to work with is the dihedral group D_{2n} given by $\langle a, b | a^2 = 1, b^n = 1, ab = b^{-1}a \rangle$, [11]. This group is non-abelian for $n \geq 3$. Every element u in RD_{2n} may be written as $u = f(b) + ag(b)$ with $f(b), g(b) \in RC_n$ where C_n is generated by b. The collection of differences of u is easy to determine. The corresponding matrix U of u has the form $\begin{pmatrix} A & B \\ B & A \end{pmatrix}$ where A is circulant and B is reverse circulant, [3]. This gives non-commutative matrices and non-commutative codes.

3 Examples, simulations and comparisons

In this section examples and simulations of the method are given and some comparisons are made with known codes. The sizes of the examples are chosen in order to compare with known examples. However there is no theoretical limit on size, the constructions are easy to perform and there is complete freedom as to choice of rows or columns to delete from a particular unit in order obtain LDPC codes with no short cycles.

The simulations compare very favourably with known examples and in some cases outstrip these.

This algebraic method for construction has other advantages such as for applications where low storage and low power are requirements. The code may be stored by an algebraic formula with few parameteres and the check matrix restored as required line-by-line without the need to store the whole structure in memory.

3.1 The examples generally

In general the examples are taken from unit-derived codes within $\mathbb{Z}_2(C_n \times C_4)$, where $\mathbb{Z}_2 = \mathbb{F}_2$ is the field of two elements.

The matrices derived are then submatrices of circulant-by-circulant matrices and are easy to program.

Assume that C_n is generated by g and C_4 is generated by h.

Every element in the group ring is then of the form: $\sum_{i=0}^{n-1}(\alpha_i g^i + h\beta_i g^i + h^2\gamma_i g^i + h^3\delta_i g^i)$, with $\alpha_i, \beta_i, \gamma_i, \delta_i \in \mathbb{Z}_2$.

3.2 (96,48) examples

These are derived from $\mathbb{Z}_2(C_{24} \times C_4)$.

The check element $v = g^{24-9} + g^{24-15} + g^{24-19} + hg^{24-3} + hg^{24-20} + h^2 g^{24-22} + h^3 g^{24-22} + h^3 g^{24-12}$ is used to define the LDPC code TFI-96-59-8.
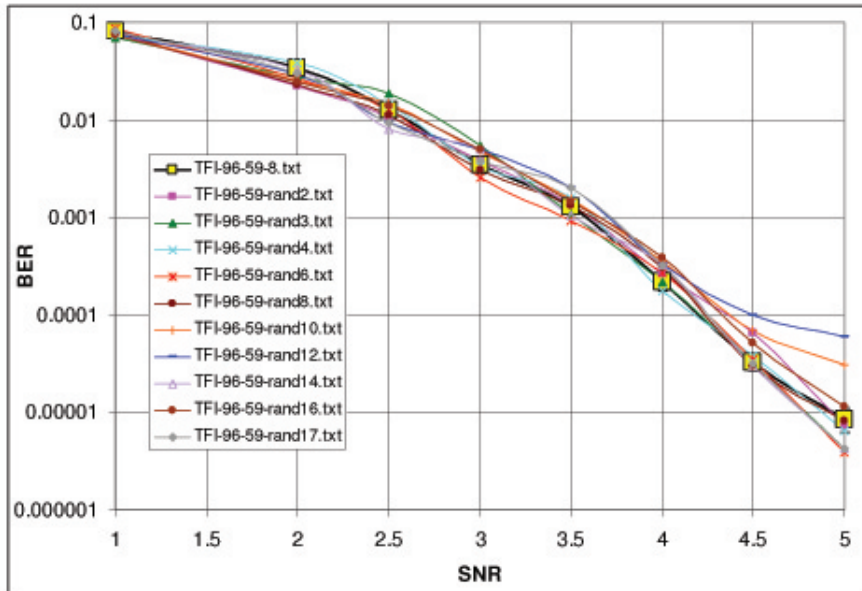
It is easy to check from Theorem 2.2 that v has no short cycles in its matrix V.

A pattern to delete half the columns from the matrix V of v is chosen to produce the rate $1/2$ code TFI-96-59-8 .

TFI-96-59-8 is compared to pseudo-random code MK-96-33-964 (size=96, rate=1/2) of MacKay [7].
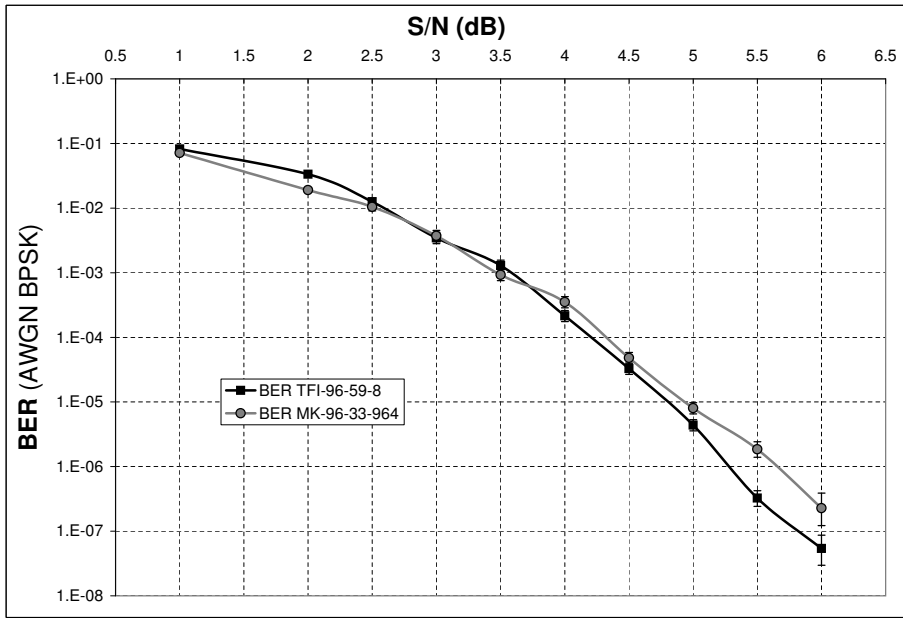
3.3   Random selection

For the above, 10 random LDPC $(96, 49)$ codes were taken from the unit v in Section 3.2 and simulated. The simulation of TFI-96-59-8, used in the previous graph where it is compared to MK-96-33-964, is included for comparison.

3.4   (504,252) example

The next example is derived from $\mathbb{Z}_2(C_{126} \times C_4)$.

$$v = g^{126-10} + g^{126-99} + hg^{126-47} + h^2(g^{126-15} + g^{126-25} + g^{126-81}) + h^3(g^{126-6} + g^{126-23} + g^{126-64}).$$

Specific column deletions are chosen from V to give the LDPC rate $1/2$ code TFI-504-91-0. The performance of TFI-504-91-18 is compared to that of PEGReg252x504 Progressive Edge Growth, Xiao-Yu Hu, IBM Zurich Research Labs.
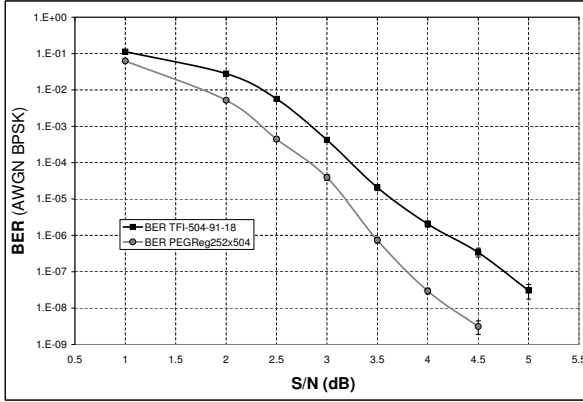
3.5   Size: 816; rate =1/2 and 3/4

Here $\mathbb{Z}_2(C_{204} \times C_4)$ is used. Set

$$v = g^{204-75} + h(g^{204-13} + g^{204-111} + g^{204-168}) + h^2(g^{204-29} + g^{204-34} + g^{204-170}) + h^3(g^{204-27} + g^{204-180}).$$

Half the columns of V are deleted in a specific manner to get the TFI-816-0p5-29-4 rate $1/2$ code.

The same v is taken and specific three quarters of the columns of V are deleted to get the $3/4$ rate $(816, 612)$ LDPC code TFI-816-0p75-29-4.

In the first graph the performances of TFI-816-0p5-29-4 and TFI-816-0p75-29-4 are compared.

In the second graph the performances of TFI-816-0p5-29-4 and MK-816-55-156, a pseudo-random rate $1/2$ code due to MacKay, [7], are compared.
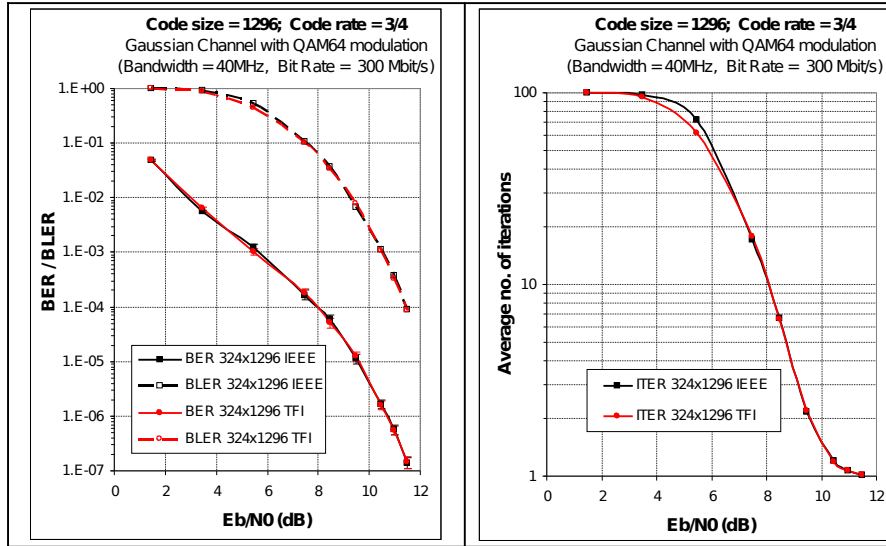
8
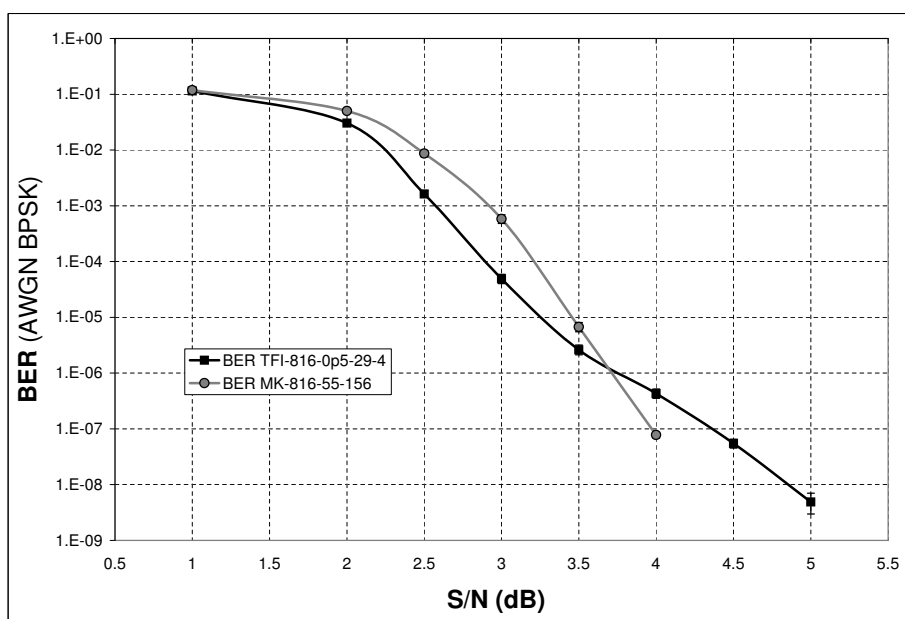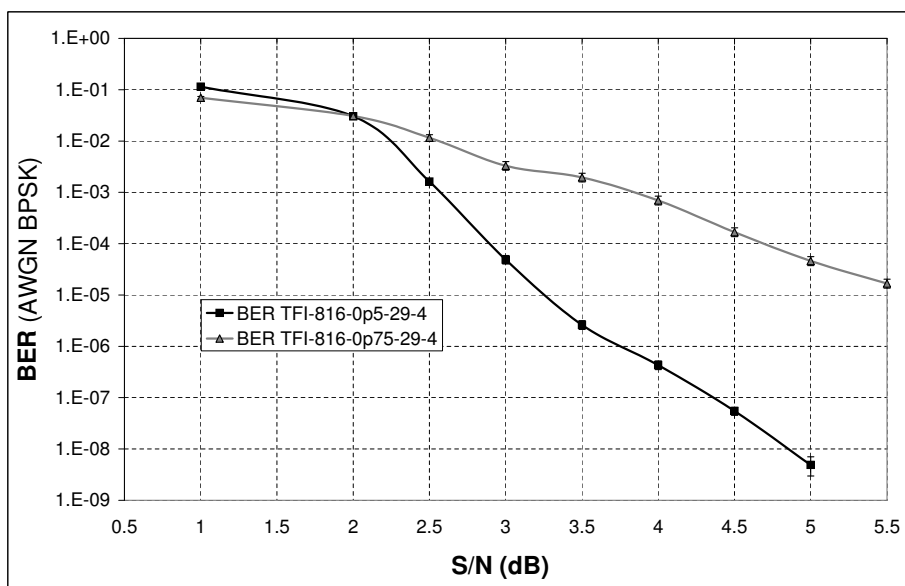
3.6 Industry Standards

Here comparisons of Bit Error Rate (BER) and Block Error Rate (BLER) performance of LDPC codes defined in the 802.11n & 802.16e standard with equivalent codes generated by present method are given.
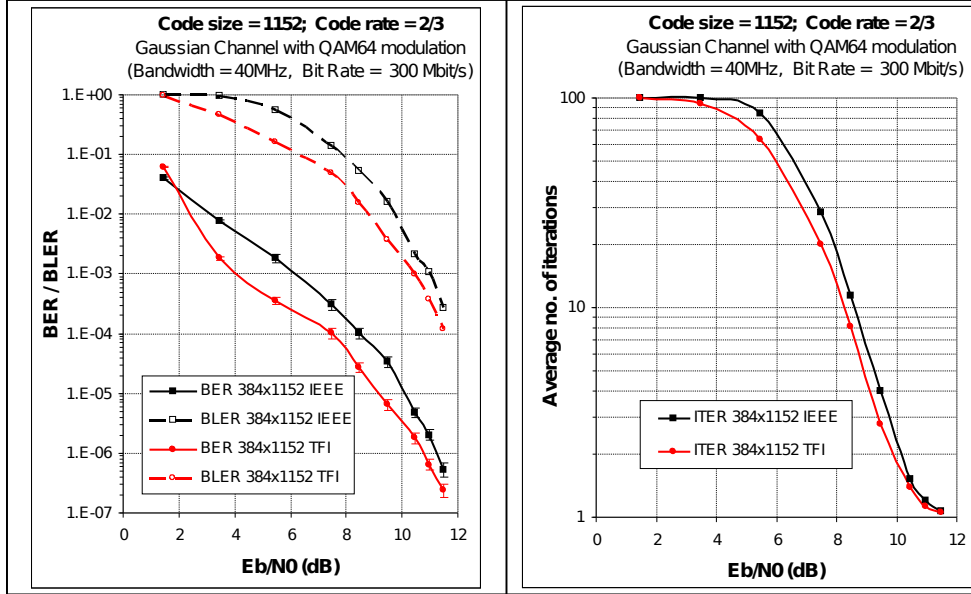
3.6.1 Case 1, 802.11n:
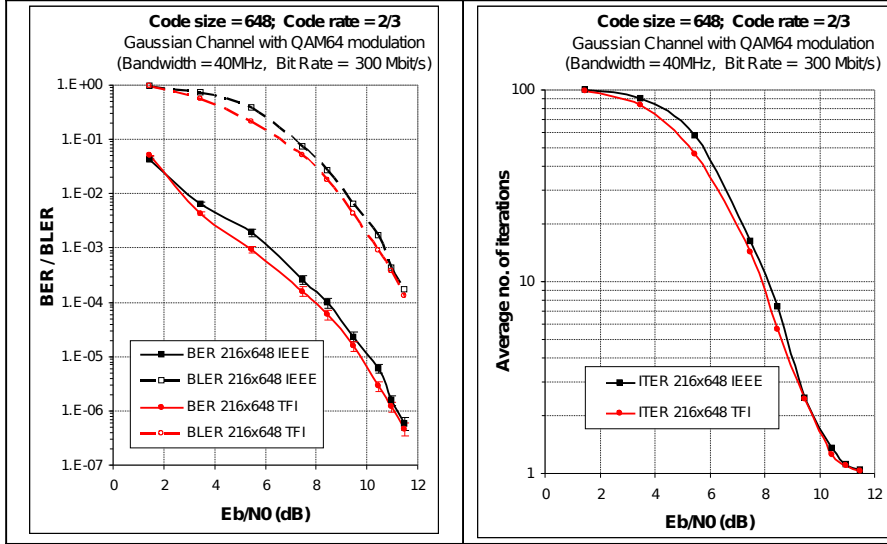
Matrix size: 216 by 648; Code size = 648; Code rate = 2/3. Matrix structure: The last 189 columns contain a 'staircase' structure which is identical as in the IEEE matrix. The remaining part was generated using the group ring algebraic algorithm which takes 15 initial parameters as input.

3.6.2 Case 2, (802.11n):

Code size = 1296; Code rate = 3/4
Gaussian Channel with QAM64 modulation
(Bandwidth = 40MHz, Bit Rate = 300 Mbit/s)

Matrix size: 324 by 1296; Code size = 1296; Code rate = 3/4. Matrix structure: The last 270 columns contain a 'staircase' structure which is identical as in the IEEE matrix. The remaining part was generated using the algebraic group ring algorithm which takes 17 initial parameters as input.

3.6.3 Case 3, (802.16e):

Matrix size: 384 by 1152; Code size = 1152; Code rate = 2/3. Matrix structure: The last 336 columns contain a 'staircase' structure which is identical as in the IEEE matrix. The remaining part was generated using the algebraic group ring algorithm which takes 17 initial parameters as input.

References

[1] R.G. Gallager, *Low-density parity-check codes*, MIT Monograph, 1963.

[2] van Lint, J.H. and Wilson, R.M., *A course in Combinatorics*, Cambridge University Press, 2001.

[3] Ted Hurley, "Group rings and rings of matrices", Inter. J. Pure & Appl. Math., 31, no.3, 2006, 319-335.

[4] Paul Hurley & Ted Hurley, "Module codes in group rings"; ISIT2007, Nice, 1981-1985, 2007.

[5] Ted Hurley, "Codes from zero-divisors and units in Group Rings", arXiv: 0710:5873.

[6] David MacKay, *Information Theory*, CUP, 2005.

[7] http://www.inference.phy.cam.ac.uk/mackay/codes/data.html

[8] D.J.C. MacKay & R.M. Neal, "Good codes based on very sparse matrices" *Crytpography and Coding: 5th IMA Conf.* , Lecture notes series in computer science, 1025 (1995), Springer, 100-111.

[9] D.J.C. MacKay & R.M. Neal, "Near Shannon limit of low density parity check codes", Electronic Letters, 32(1996), 1645-1646.

[10] David J C MacKay, "Good Error-Correcting Codes based on Very Sparse Matrices", IEEE Trans. Inform. Theory, 45(1999), 399-431.

[11] César Milies & Sudarshan Sehgal, *An introduction to Group Rings*, Klumer, 2002.

[12] O. MILENKOVIC, I. DJORDJEVIC, AND B. VASIC, "Block-circulant low-density parity-check codes for optical communication systems", IEEE Journal of Selected Topics in Quantum Electronics, 10 (2004), pp. 294–299.

[13] H. TAN, JUN XU, YU KOU, SHU LIN, AND KHALED A. S. ABDEL-GHAFFAR, "On algebraic construction of Gallager and circulant low-density parity-check codes", IEEE Trans. Inform. Theory, 50 (2004), pp. 1269–1279.

[14] R. Tanner, "A recursive approach to low complexity codes", IEEE Trans. Inform. Theory, 27 (1981), 533–547.

[15] R.M. TANNER, D. SRIDHARA, A. SRIDHARAN, T.E. FUJA, AND DANIEL J. COSTELLO JR., "LDPC block and convolutional codes based on circulant matrices", IEEE Trans. Inform. Theory, (2004), pp. 2966–2984.